

Predicción del crecimiento de un cultivo de *Coffea arabica* por medio de inteligencia artificial
para la agricultura de precisión implementado en el dispositivo Farmbot

Hernán David Sánchez Restrepo

Trabajo de Grado para Optar al Título de Ingeniero Mecánico

Director

Octavio Andrés Gonzáles Estrada

PhD en Ingeniería Mecánica y Materiales

Codirector

Cristian Andrés Hernández Salazar

Ingeniero Mecánico

Universidad Industrial de Santander

Facultad de Ingeniería Físico-Mecánicas

Escuela de Ingeniería Mecánica

Bucaramanga

2024

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a Cristian Hernández y Andrés González, cuya asistencia y apoyo fueron cruciales para el desarrollo de este proyecto. La experiencia y conocimientos compartidos por ambos no solo enriquecieron significativamente este trabajo, sino que también facilitaron la superación de numerosos desafíos técnicos y metodológicos a lo largo del proceso investigativo.

Tabla de Contenido

	Pág.
Introducción	11
1. Objetivos.....	14
1.1 Objetivo General.....	14
1.2 Objetivos Específicos.....	14
2. Materiales y métodos	15
2.1 Marco Referencial.....	15
2.1.1 Farmbot:.....	15
2.1.2 Deep Learning.....	15
2.1.3 PlantCV	16
2.1.4 Naïve Bayes	16
2.1.5 U2net.....	17
2.1.6 Salient object detection	19
2.2 Métodos.....	19
2.2.1 Preprocesamiento.....	21
2.2.2 Predicción del crecimiento de la planta	29
3. Resultados.....	33
3.1 PlantCV.....	33
3.1.1 Segmentación dual HSV/LAB.....	34
3.1.2 Segmentación por Naive Bayes	39
3.2 Red neuronal U2-net.....	41
3.3 Análisis de espectro cromático en RGB	44

3.4 Predicción del crecimiento de la planta 45

4. Conclusiones 48

Referencias Bibliográficas 50

Lista de Tablas

	Pág.
Tabla 1 <i>Arquitectura de red U2</i>	19
Tabla 2 <i>Valores RGB del pixel de la planta seleccionado</i>	25
Tabla 3 <i>Valores RGB del pixel para el fondo</i>	26
Tabla 4 <i>Hiperparámetros de la red LSTM propuesta</i>	32
Tabla 5 <i>Validación cruzada por k-fold</i>	46

Lista de Figuras

	Pág.
Figura 1 <i>Bloque residual U (RSU). C_{inv} C_{out} representa el canal de entrada y salida, M el número de canales de las capas intermedias y L el número de capas en el codificador. H y W el tamaño de la imagen. Fuente. Elaboración propia.</i>	17
Figura 2 <i>Metodología del flujo de trabajo del proyecto. Fuente. Elaboración propia</i>	20
Figura 3 <i>Canales de selección para limpieza de imagen HSV/LAB Fuente. Elaboración propia.</i>	21
Figura 4 <i>Visualización cartesiana de la paleta cromática. Fuente. Elaboración propia</i>	22
Figura 5 <i>Separación de pixeles por paleta cromática. Elaboración. Propia.</i>	23
Figura 6 <i>Máscara por segmentación dual. Fuente. Elaboración propia</i>	24
Figura 7 <i>Pixel inspector de ImageJ. Fuente. Elaboración propia</i>	25
Figura 8. <i>Imagen segmentada mediante Naive Bayes. Fuente. Elaboración propia</i>	27
Figura 9 <i>Imagen a segmentar. Fuente. Dataset de proyecto en conjunto.</i>	28
Figura 10 <i>Imagen segmentada por U2-net. Fuente. Elaboración propia</i>	29
Figura 11 <i>Vista expandida de red LSTM</i>	30
Figura 12 <i>Arquitectura interna de la red LSTM. Fuente. Elaboración propia.</i>	31
Figura 13 <i>Planta para 01-1 el 10 de mayo del 2023. Fuente. Elaboración propia.</i>	33
Figura 14 <i>Máscara de Planta 01-1 el 10 de mayo del 2023 segmentada de forma dual. Fuente. Elaboración propia</i>	34
Figura 15 <i>Máscara final por segmentación dual por HSV/LAB. Fuente: Elaboración propia</i>	35
Figura 16 <i>Planta 01-1 el 10 mayo del 2023 segmentada de forma dual. Fuente. Elaboración propia</i>	36

Figura 17 *Morfología de la planta 01-1 el 10 de mayo de 2023. Fuente. Elaboración propia....* 37

Figura 18 *Planta 01-1 el 14 de octubre del 2023 segmentada de forma dual. Fuente. Elaboración propia.....* 38

Figura 19 *Morfología de la planta 01-1 el 14 octubre del 2023. Fuente. Elaboración propia....* 39

Figura 20 *Planta 01-1 en mayo del 2023 segmentada por Naive Bayes. Fuente. Elaboración propia.....* 40

Figura 21 *Planta 01-1 en octubre del 2023. Fuente. Elaboración propia.* 41

Figura 22 *Planta 01-1 en mayo segmentada por U2net. Fuente. Elaboración propia.* 42

Figura 23 *Planta 01-1 en octubre. Fuente. Elaboración propia.* 42

Figura 24 *Comparación de segmentaciones: Naive Bayes, dual HSV/LAB y U2 net respectivamente. Fuente. Elaboración propia.....* 43

Figura 25 *Resultados de segmentación de U2 net por zonas. Fuente. Elaboración propia.....* 43

Figura 26 *Análisis cromático en RGB de las imágenes sin segmentar. Fuente. Elaboración propia.....* 44

Figura 27 *Análisis cromático en RGB de las imágenes segmentadas. Fuente. Elaboración propia.....* 45

Figura 28 *Error cuadrático medio vs número de épocas en entrenamiento. Fuente. Elaboración propia.....* 46

Figura 29 *Predicción de crecimiento del diámetro de la planta por zonas. (a)Zona A, (b)Zona B, (c)Zona C, (d)Zona D. Fuente. Elaboración propia.....* 47

Resumen

Título: Predicción del crecimiento de un cultivo de *Coffea arabica* por medio de inteligencia artificial para la agricultura de precisión implementado en el dispositivo Farmbot

Autor: Hernán David Sánchez Restrepo **

Palabras Clave: Deep Learning, Farmbot, coffea arabica, convolutional neural networks, precision agriculture.

Descripción:

La automatización es el futuro y el objetivo en todos los ámbitos de ingeniería, y los cultivos agrícolas no son ajenos al mismo. Por ello, desde tiempo atrás, se ha estado desarrollando la agricultura de precisión; uno de sus retos es la segmentación de imágenes, ya que para tener un monitoreo permanente es necesario la limpieza y clasificación de las imágenes proporcionadas por la sensorica que supervisa al cultivo. Esta investigación propone una metodología para la segmentación de la planta de coffea arabica. Las imágenes utilizadas provienen de un conjunto de datos creado en un proyecto anterior con la ayuda de un dispositivo Farmbot. Se desarrolló una red neuronal utilizando aprendizaje profundo con el modelo U2-Net. Se introdujo una base de datos de 5427 imágenes en la que la planta de café está rodeada de vegetación y maleza. Estos elementos se presentan como objetos destacados que dificultan la segmentación, ya que se interpretan como ruido al procesar las imágenes. Sin embargo, esta característica también fortalece la red neuronal, ya que desarrolla la capacidad de distinguir entre la vegetación común y la planta de *coffea arabica*. Los resultados de la segmentación muestran una precisión del 67%.

** Trabajo de grado

Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería Mecánica. Director: Octavio Andrés Gonzales Estrada. PhD en Ingeniería Mecánica y Materiales. Codirector: Cristián Andrés Hernández Salazar. Título Ingeniero Mecánico, Hernán David Sánchez Restrepo Título Ingeniero Mecánico.

Con los datos obtenidos por las imágenes segmentadas y las condiciones de su entorno, se realizó una predicción del crecimiento de las plantas, obteniendo una función del crecimiento muy similar a la real.

Abstract

Title: Prediction of the growth of a *Coffea arabica* crop by artificial intelligence for precision agriculture implemented in the Farmbot device.

Author: Hernán David Sánchez Restrepo**

Key Words: Deep Learning, Farmbot, coffea arabica, convolutional neural networks, precision agriculture.

Description: Automation is the future and the goal in all engineering fields, and agricultural crops are no stranger to it. For this reason, precision agriculture has been developing for some time now; one of its challenges is image segmentation, since in order to have a permanent monitoring it is necessary to clean and classify the images provided by the sensor system that supervises the crop. This research proposes a methodology for the segmentation of the coffea arabica plant. The images used come from a dataset created in a previous project with the help of a Farmbot device. A neural network was developed using deep learning with the U2-Net model. A database of 5427 images was introduced in which the coffee plant is surrounded by vegetation and weeds. These elements are presented as prominent objects that make segmentation difficult, as they are interpreted as noise when processing the images. However, this feature also strengthens the neural network, as it develops the ability to distinguish between common vegetation and the coffea arabica plant. The segmentation results show an accuracy of 67%. With the data obtained from the segmented images and the conditions of their environment, a prediction of plant growth was made, obtaining a growth function very similar to the real one.

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería Mecánica. Director: Octavio Andrés Gonzales Estrada. PhD en Ingeniería Mecánica y Materiales. Codirector: Cristián Andrés Hernández Salazar. Título Ingeniero Mecánico, Hernán David Sánchez Restrepo Título Ingeniero Mecánico.

Introducción

El avance de la ingeniería ha sido uno de los principales impulsores de los procesos industriales, desde la primera revolución industrial se ha hecho evidente el desarrollo y crecimiento económico ocasionados por la adquisición de nuevas tecnologías para satisfacer las demandas sociales(Didier, 2024). Este progreso tecnológico ha impulsado no solo la reducción de los tiempos de producción, sino también los costos operativos, lo que conlleva a un aumento notable en la productividad. La reducción de costos operativos se logra tanto por un mejor aprovechamiento de materias primas, gracias al uso de maquinaria para su selección(Cui et al., 2021), como por la gestión del personal de trabajo, ya que la rotación y/o escasez de empleados son factores que introducen una inestabilidad indeseada en el proceso productivo industrial(Abicht et al., 2024).

Una solución a este problema es el progreso en la automatización industrial, donde el uso del internet industrial de las cosas (IIoT, por sus siglas en ingles) en conjunto con la inteligencia artificial, elimina dicho factor variacional y establece una producción estable (Madakam et al., 2019). El IIoT a través de sensores, instrumentos de medición, actuadores electrónicos y robótica, nos permite acceder a un flujo de información para monitorear, analizar y predecir el comportamiento de los sistemas de producción de manera remota, las 24 horas del día. Así mismo, facilita del control de todo tipo sistemas industriales, ya sea en procesos textiles, farmacéuticos, automovilísticos, transformación de alimentos o incluso de agricultura(Sundari et al., 2021). Esto resulta en una mejora en la eficiencia de los procesos de automatización industrial. Sin embargo, esto no es el objetivo final, pues el fin yace no en el grado de automatización, sino en la evolución, optimización y desarrollo autónomo del sector productivo

de las empresas. Esto se conoce como “industrias inteligentes”(Kusiak, 2018), donde la inteligencia artificial por medio de algoritmos de machine learning, procesa el flujo de información otorgándoles a los sistemas industriales un valor agregado, donde resalta su capacidad mejorada de percepción, cognición, organización, aprendizaje y ejecución de decisiones propias para la toma de decisiones(Leng et al., 2024).

En Colombia, el café representa una de las principales fuentes de ingresos de la región cafetera, solo en Santander se encuentran sembradas 53.017 hectáreas en 75 municipios(Federación Nacional de Cafeteros de Colombia, 2023), un aumento en su producción es esencial tanto en materia económica como en términos de seguridad alimentaria, pues para el 2050, se estima que la producción alimentaria tendrá que aumentar entre un 25% y un 75% respecto a los niveles de producción registrados en el 2017(Hunter et al., 2017). Con el fin de satisfacer esta necesidad se inició este proyecto de investigación el cual está enfocado en impulsar la tecnificación de la producción industrial del café colombiano, específicamente la producción de *Coffea arabica*. La agricultura de precisión ha evolucionado significativamente en los últimos años gracias a la integración tecnologías emergentes, como la inteligencia artificial (IA) con la cual se ha podido identificar fenotípicamente el tipo de planta que se está sembrando(Quan et al., 2023) y la automatización con la cual ha sido posible reducir costos en mano de obra(SgROI, 2023), fertilizante y agua por medio del monitoreo de los cultivos(Lee et al., 2023), incluso se ha logrado detectar hojas de plantas que presentan secciones dañadas o enfermas(Abuhayi & Mossa, 2023). De esta forma, la incorporación de tecnologías de inteligencia artificial en sistemas agrícolas industriales como el dispositivo Farmbot, se perfila como una estrategia innovadora en respuesta a los retos actuales de la agricultura de precisión. Es por esta razón que este proyecto propone una metodología para predecir el crecimiento de cultivos de café usando

inteligencia artificial para impulsar el desarrollo y aplicabilidad de la automatización en el sector agrícola. Se usó una base de datos compuesta por imágenes de un cultivo controlado de *Coffea arabica*, las cuales fueron registradas por el dispositivo Farmbot durante un periodo de 6 meses. Este registro fotográfico diario forma parte de un proyecto macro que envuelve ambos trabajos de investigación. A las imágenes se les aplicó una segmentación inicial con el fin de obtener mejores resultados a la hora de predecir el crecimiento de la planta por medio de inteligencia artificial. Este enfoque tecnológico tiene la capacidad de monitorear y pronosticar el desarrollo del cultivo de manera eficaz, lo cual allana el camino hacia la automatización completa en la industria agrícola.

1. Objetivos

1.1 Objetivo General

Predecir el crecimiento de un cultivo de *Coffea arabica* a través de la construcción de un algoritmo de inteligencia artificial usando Deep Learning para brindar un aporte a la automatización en la agricultura de precisión.

1.2 Objetivos Específicos

Desarrollar un algoritmo de Deep Learning utilizando redes convolucionales que, al procesar cada imagen capturada por el Farmbot, sea capaz de identificar y segmentar la planta de *Coffea arabica*, diferenciándola de su entorno mediante la utilización de la librería PlantCV, para que posteriormente el Farmbot ejecute el algoritmo de forma autónoma.

Ajustar el modelo para capacitarlo en la predicción del crecimiento futuro de una planta de café utilizando datos de los primeros meses de entrenamiento en forma de imágenes, además de variables físicas como su temperatura, humedad y nutrientes de crecimiento. Permitiendo así que el algoritmo genere nuevas representaciones visuales del desarrollo de la planta de café.

Evaluar la eficacia del algoritmo desarrollado mediante validación cruzada y la validación en un conjunto de datos de prueba independiente, con el propósito de medir con precisión su capacidad para trabajar con imágenes reales y generar predicciones precisas sobre el crecimiento de las plantas de *Coffea arabica*.

2. Materiales y métodos

2.1 Marco Referencial

En el marco referencial que se presenta a continuación, se analizan los aspectos fundamentales que se derivan de los antecedentes y referentes teóricos. A partir de esto, se presenta un resumen de los conceptos principales que son relevantes para el desarrollo del proyecto.

2.1.1 Farmbot:

FarmBot es un proyecto de código abierto que tiene como objetivo ayudar a las familias, los estudiantes y los científicos a cuidar sus propios huertos sin tener que hacer mucho. El robot se parece a una impresora gigante controlada por Raspberry Pi que puede moverse por toda la superficie del huerto. Tiene varios cabezales que cambian según quieras plantar semillas, regar o matar malas hierbas. Una aplicación que puedes usar desde tu móvil, tableta o PC para controlar todo. Aquí, puedes diseñar tu huerto y especificar dónde quieres plantar cada tipo de semilla, y FarmBot las plantará en el lugar adecuado. Además, tiene una cámara para tomar fotos(Nugraha et al., 2020).

2.1.2 Deep Learning

El Deep Learning es un subconjunto del aprendizaje automático, que consiste esencialmente en una red neuronal con tres o más capas. Estas redes neuronales intentan simular el comportamiento del cerebro humano -aunque están lejos de igualar su capacidad- permitiéndole "aprender" a partir de grandes cantidades de datos. Aunque una red neuronal con una sola capa puede hacer predicciones aproximadas, las capas ocultas adicionales pueden ayudar a optimizar y refinar la precisión, sin embargo, tienen un costo computacional alto(IBM, 2023).

2.1.3 PlantCV

PlantCV es un paquete de software de análisis de imágenes de código abierto para el fenotipado de plantas. PlantCV soporta actualmente el análisis de imágenes en color RGB estándar (también conocidas como "VIS"), imágenes en escala de grises estándar (por ejemplo, infrarrojo cercano, "NIR"), imágenes infrarrojas térmicas, imágenes en escala de grises de sistemas de imágenes de fluorescencia de clorofila ("PSII") e imágenes hiperespectrales ("ENVI")(Berry et al., 2018). Actualmente recibe apoyo económico por parte del departamento de agricultura de estados unidos (USDA) para impulsar el desarrollo del software.

2.1.4 Naïve Bayes

El clasificador Naïve Bayes es una técnica de aprendizaje supervisado que se basa en el teorema de probabilidad de Naïve Bayes(Ahmad et al., 2022). Una de sus ventajas es que requiere poca información de entrenamiento y es computacionalmente eficiente(Sano et al., 2023). El teorema de probabilidad está definido tal que:

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)} \quad (1)$$

Donde:

$P(y | x_1, \dots, x_n)$: Es la probabilidad posterior de la clase y dadas las características x_1 hasta x_n .

$P(x_1, \dots, x_n | y)$: Es la probabilidad de que las características x_1 hasta x_n pertenezcan a la clase y .

$P(y)$: Es la probabilidad previa de la clase y .

$P(x_1, \dots, x_n)$: Es la probabilidad total de los datos.

El clasificador actúa bajo la suposición de que cada característica es independiente entre sí. Este supuesto de independencia es lo que le da el nombre "naive" o ingenuo, ya que, en la realidad, las características suelen estar relacionadas. Según esta suposición tenemos que:

$$P(y | x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i | y)}{P(x_1, \dots, x_n)} \quad (2)$$

Donde:

$\prod_{i=1}^n P(x_i | y)$: Es el producto de las probabilidades condicionales de cada característica x_i , dado que la muestra es de la clase y .

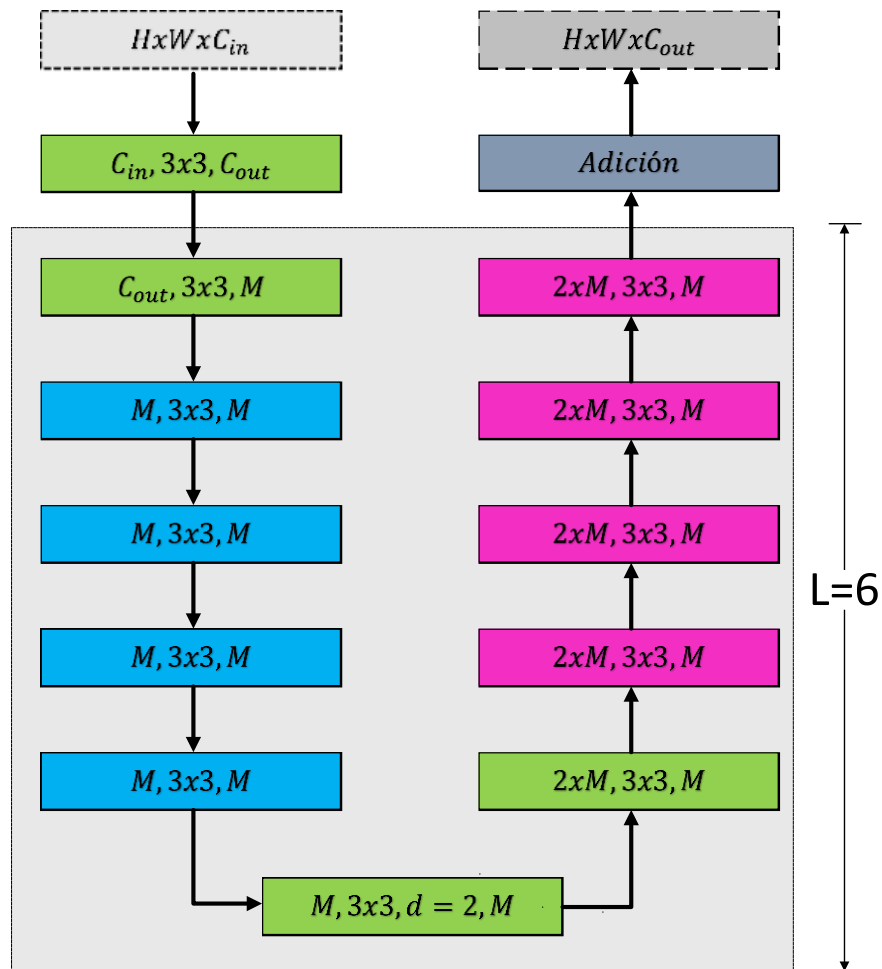
2.1.5 U2net

U2net es una arquitectura de redes neuronales que se especializa en la detección de objetos salientes (SOD) por medio del deep learning (Qin et al., 2020). Su arquitectura consiste en dos niveles de U-estructuras anidadas y se constituye en cada fase de bloques residuales en forma de “U”, como se puede observar en la Figura 1.

Figura 1

Bloque residual U (RSU). C_{inv} C_{out} representa el canal de entrada y salida, M el número de

canales de las capas intermedias y L el número de capas en el codificador. H y W el tamaño de la imagen. Fuente. Elaboración propia.



Esta arquitectura tiene la ventaja de que se puede entrenar desde cero sin utilizar los backbones de tareas de clasificación de imágenes (Qin et al., 2020). El problema con el uso de backbones preentrenados es que pueden no ser óptimos para tareas específicas fuera de la clasificación de imágenes. Esto se debe a que están diseñados para reconocer y clasificar imágenes en categorías predefinidas, lo que puede no ser completamente alineado con los requisitos de otras tareas como la detección de objetos salientes. Por lo tanto, entrenar una red desde cero, sin depender de estos backbones, permite una mejor adaptación a las necesidades específicas de la tarea a realizar, aunque esto puede requerir más datos y tiempo de

entrenamiento. La arquitectura detallada se puede observar en la Tabla 1. En_i y De_i representan la etapa de codificador y decodificador. I , M , O indican el número de canales de entrada (C_{in}), canales intermedios y canales de salida (C_{out}) respectivamente.

Tabla 1

Arquitectura de red U2.

Etapas										
En_1	En_2	En_3	En_4	En_5	En_6	De_5	De_4	De_3	De_2	De_1
RSU-7	RSU-6	RSU-5	RSU-4	RSU-4F	RSU-4F	RSU-4F	RSU-4	RSU-5	RSU-6	RSU-7
I:3	I:64	I:128	I:256	I:512	I:512	I:1024	I:1024	I:512	I:256	I:128
M:32	M:32	M:64	M:128	M:256	M:256	M:256	M:128	M:64	M:32	M:16
O:64	O:128	O:256	O:512	O:512	O:512	O:512	O:256	O:128	O:64	O:64

Nota: La información de la arquitectura de la tabla fue tomada de: “U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection”, por X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand Oct. 2020.

2.1.6 Salient object detection

Salient object detection (SOD), es una técnica que se encarga de obtener de una imagen el objeto que más resalte y eliminar o borrar todo lo demás como si fuera parte del fondo. Se usa comúnmente en la segmentación de imágenes y el seguimiento visual. Funciona a través de inteligencia artificial, entrenándose por medio Deep Learning(Qin et al., 2020).

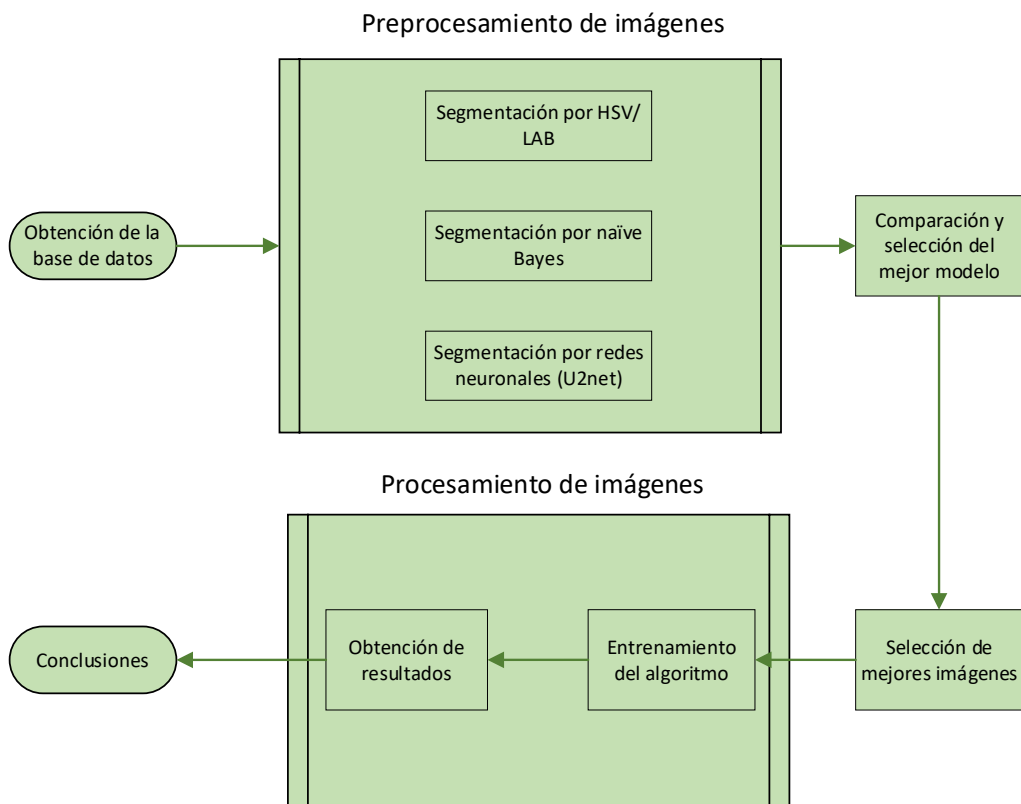
2.2 Métodos

La metodología parte de un proyecto macro en el cual se trabajó en conjunto con otros profesionales. Obteniendo una base de datos conformada por 5427 imágenes de un terreno de plántulas de *Coffea arabica* controlado a través de un dispositivo Farmbot, que registró su

crecimiento por un periodo de 6 meses. El terreno se dividió en cuatro zonas, A, B, C, D. La zona A es la zona de control, la zona B se le suministró agua, la zona C fertilizante, y a la D tanto agua como abono. Los datos fotográficos obtenidos de la zona experimental se usaron para el preprocesamiento a través de tres algoritmos diferentes de segmentación y se escogió el mejor. Posteriormente se generó una predicción del crecimiento de la planta y finalmente una evaluación de los resultados con las imágenes reales. En la Figura 2 se puede observar el diagrama de flujo de trabajo del proyecto.

Figura 2

Metodología del flujo de trabajo del proyecto. Fuente. Elaboración propia



2.2.1 Preprocesamiento

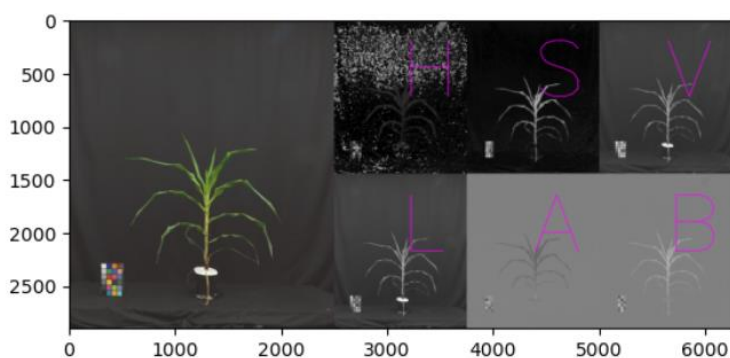
Para el preprocesamiento se utilizaron tres tipos de segmentación: El primero consistió en un análisis de imágenes utilizando modelos de color HSV/LAB, mientras que el segundo método se basó en el análisis de imágenes de fluorescencia clorofílica. Ambos métodos son parte de PlantCV, una biblioteca especializada de Python diseñada específicamente para el trabajo con plantas. El tercer método empleó un segmentador generalizado basado en redes neuronales, conocido como U2net, que ha demostrado ser el más eficaz para este tipo de tareas.

2.2.1.1 Segmentación por medio del modelo HSV/LAB

Para esta primera segmentación se usó la librería PlantCV, específicamente el módulo de HSV/LAB en donde se transforma el espacio de colores de RGB a HSV o a LAB y se diferencia la planta respecto a su entorno por medio de diferentes canales de colores, es uno de los métodos más comunes, se le conoce como umbralización. Este preprocesamiento se hace de manera manual como se observa en la Figura 3

Figura 3

Canales de selección para limpieza de imagen HSV/LAB Fuente. Elaboración propia.

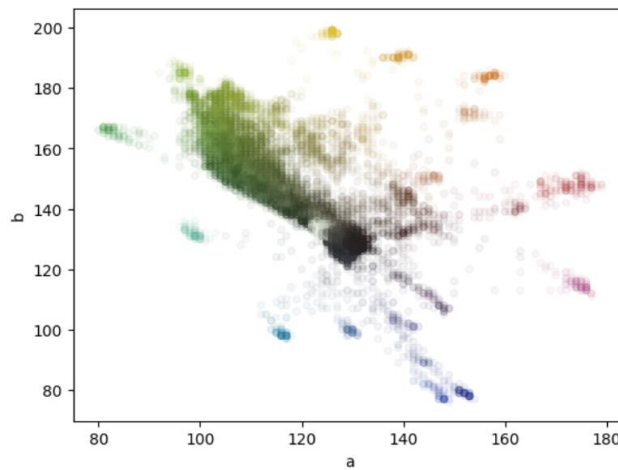


Este tipo de preprocesamiento se usa comúnmente cuando la planta sobresale de su entorno por una diferencia de colores fácilmente visible. La umbralización dual funciona de la siguiente forma: una vez identificados dos canales donde la planta resalte más de su entorno, en

este caso a y b, se crea una visualización cartesiana de la paleta cromática, como se puede observar en la Figura 4.

Figura 4

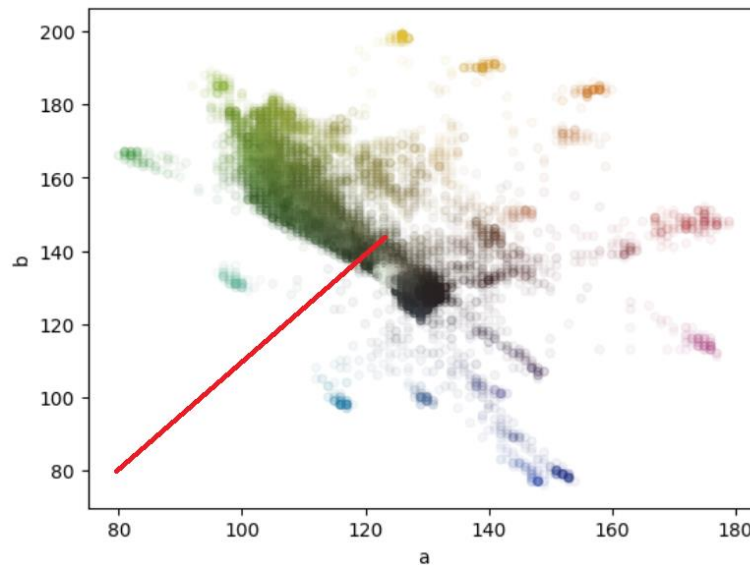
Visualización cartesiana de la paleta cromática. Fuente. Elaboración propia



Una vez creado el diagrama se seleccionan dos puntos en coordenadas cartesianas (x, y) del gráfico. Esto permite crear una línea que separe los pixeles verdes, propios de la planta, de todo lo demás, como se puede apreciar en la Figura 5.

Figura 5

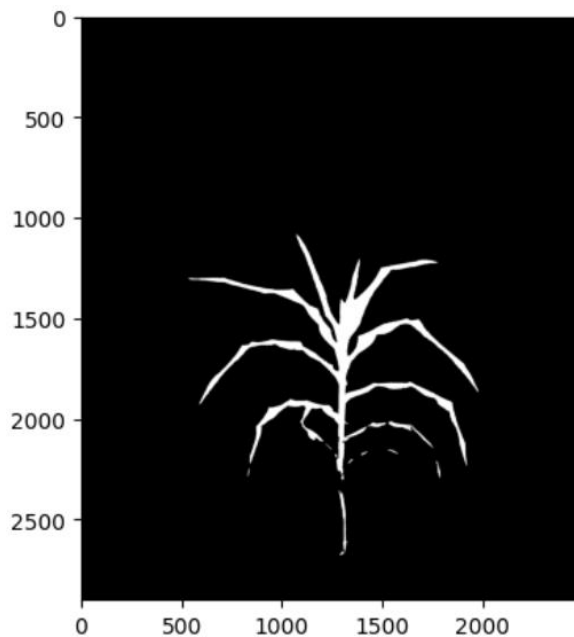
Separación de píxeles por paleta cromática. Elaboración. Propia.



Los píxeles de colores que estén por “encima” de esta línea se convertirán en un valor de 255, representado por el color blanco y todo lo demás se transformará en 0, es decir, en negro. La Figura 6 se muestra el resultado de la segmentación dual. A esta imagen binaria se le conoce como “mask” o máscara y puede ser reutilizada para entrenar otros algoritmos.

Figura 6

Máscara por segmentación dual. Fuente. Elaboración propia



Una vez obtenida esta máscara, se usa un script para rehacer la segmentación en todo el repositorio de imágenes de manera automática, pues todas las imágenes presentan el mismo espectro cromático.

2.2.1.2 Segmentación por medio del algoritmo Naïve Bayes

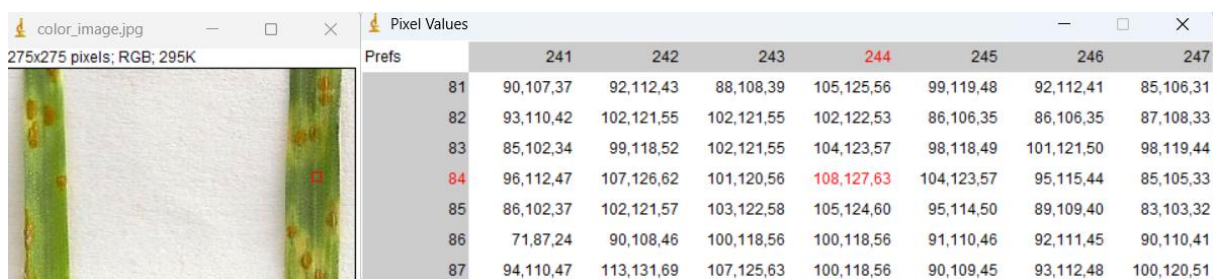
El algoritmo de clasificación de Naïve Bayes también hace parte de la librería PlantCV, y funciona de la siguiente manera: Genera una imagen binaria (etiquetada) a partir de una imagen a color utilizando la información estadística aprendida de un conjunto de entrenamiento. Del conjunto de entrenamiento, calculamos la función de densidad de probabilidad (FDP) para las Variables Aleatorias Planta (P) y Fondo (B), cada una con las Variables Aleatorias Tono (H), Saturación (S) y Valor (V) (canales de color). Con estas FDP, determinamos la probabilidad conjunta de que un píxel pertenezca a la Variable Aleatoria P o B aplicando el Teorema de Bayes bajo la suposición ingenua de que las Variables Aleatorias son independientes (por

conveniencia). Finalmente, los píxeles se etiquetan como parte de la planta (255) o del fondo (0), basándose en cuál de las dos probabilidades es mayor.

Con el fin de obtener datos de entrenamiento para el algoritmo se utiliza un software de recolección de datos de RGB llamado ImageJ. ImageJ tiene una herramienta llamada “Pixel inspector” que recoge los valores de cada canal, Red, Green, Blue, en un radio alrededor del pixel central seleccionado dentro de la imagen, como se puede ver en la Figura 7.

Figura 7

Pixel inspector de ImageJ. Fuente. Elaboración propia.



Se selecciona un pixel que contenga a la planta y se registran los datos. Los valores RGB del pixel seleccionado correspondiente a la planta se pueden ver en la Tabla 2.

Tabla 2

Valores RGB del pixel de la planta seleccionado.

Valores RGB de la planta						
90,107,37	92,112,43	88,108,39	105,125,56	99,119,48	92,112,41	85,106,31
93,110,42	102,121,55	102,121,55	102,122,53	86,106,35	86,106,35	87,108,33
85,102,34	99,118,52	102,121,55	104,123,57	98,118,49	101,121,50	98,119,44
96,112,47	107,126,62	101,120,56	108,127,63	104,123,57	95,115,44	85,105,33
86,102,37	102,121,57	103,122,58	105,124,60	95,114,50	89,109,40	83,103,32
71,87,24	90,108,46	100,118,56	100,118,56	91,110,46	92,111,45	90,110,41

94,110,47 113,131,69 107,125,63 100,118,56 90,109,45 93,112,48 100,120,51

Se repite el mismo procedimiento seleccionando ahora el fondo y se guarda registro. Los valores seleccionados se pueden ver en la Tabla 3.

Tabla 3

Valores RGB del pixel para el fondo.

Valores RGB del fondo						
230,229,224	228,227,222	224,223,218	220,219,214	220,219,214	225,224,219	232,231,226
239,238,233	235,234,229	237,236,231	239,238,233	239,238,233	239,238,233	237,236,231
234,233,228	235,234,229	233,232,227	230,229,224	227,226,221	230,229,224	236,235,230
244,243,238	244,243,238	243,242,237	240,239,234	239,238,233	243,242,237	246,245,240
234,233,228	234,233,228	238,237,232	236,235,230	238,237,232	240,239,234	242,241,236
240,239,234	240,239,234	239,238,233	233,232,227	232,231,226	235,234,229	238,237,232
226,225,220	231,230,225	237,236,231	236,235,230	234,233,228	230,229,224	229,228,223

Se almacenan los datos en un archivo .txt que tiene de encabezado cada tipo de clase antecedido de un #. En este caso se llamará #plant a los valores RGB correspondientes a la planta y #background a los valores RGB correspondientes al fondo. Y se guarda el archivo con el nombre “pixel_inspector_rgb_values.txt”.

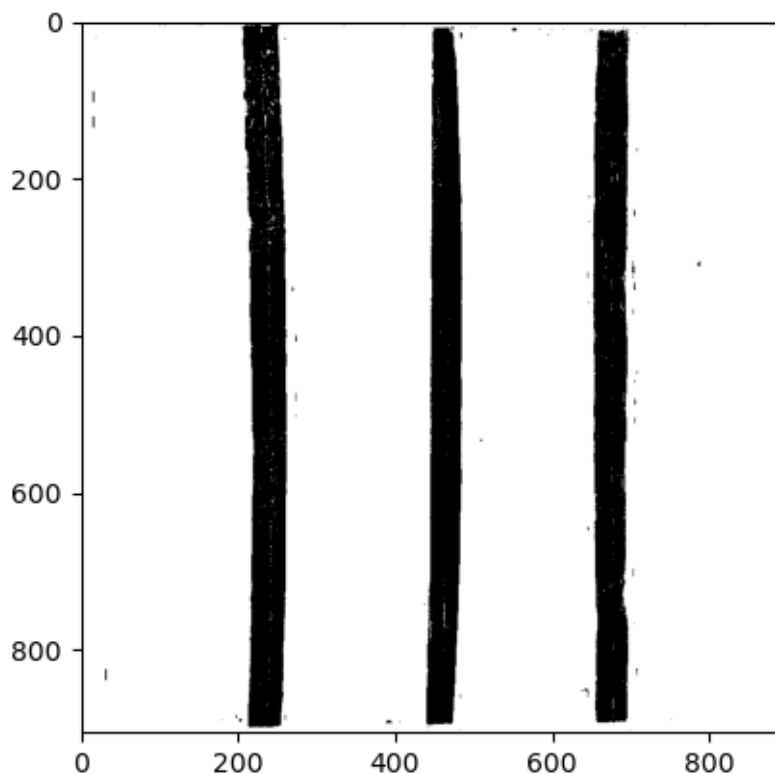
Una vez obtenidos los datos de entrenamiento se entrenará el algoritmo de Naive Bayes con la función plantcv-train.py. Esta función creará un archivo que incluirá una línea para cada uno de los canales de color (H, L y V) correspondientes a cada categoría. Las dos primeras columnas indicarán la clase y el tipo de canal, respectivamente. Las siguientes 256 columnas

presentarán el valor de probabilidad obtenido de las funciones de densidad de probabilidad para cada nivel de intensidad que se puede observar en una imagen de 8 bits, que van de 0 a 255.

Luego se procederá a segmentar la imagen usando la función `pcv.naive_bayes_classifier` usando la imagen a segmentar y el archivo con la función de densidad de probabilidad. Con esto obtendremos la máscara como se puede observar en Figura 8.

Figura 8.

Imagen segmentada mediante Naive Bayes. Fuente. Elaboración propia.



2.2.1.3 Segmentación por medio de U2-net

El uso de redes neuronales es bastante sencillo, su única dificultad parte de tener la suficiente información para someterla a un buen entrenamiento. Para ello, se recomienda usar el dataset DUTS-TR que está compuesto por 10.533 imágenes, cada una con su respectiva máscara.

DUST-TR es la base de datos más grande y frecuentemente usada para la detección saliente de objetos(Qin et al., 2020). Vale la pena aclarar que el dataset está compuesto por toda clase de

objetos, no solo por plantas. Con el fin de mejorar el entrenamiento, se doblan las imágenes reflejándolas horizontalmente y así duplicar la información de entrenamiento. Una vez entrenada la red neuronal se inicia la segmentación simplemente cargando las imágenes. La Figura 10 muestra la imagen a segmentar.

Figura 9

Imagen a segmentar. Fuente. Dataset de proyecto en conjunto.



Una vez seleccionada la imagen, se ejecuta el código de evaluación especificando la dirección de la imagen o el conjunto de imágenes a segmentar. La Figura 10 muestra el resultado de la segmentación por U2-net.

Figura 10

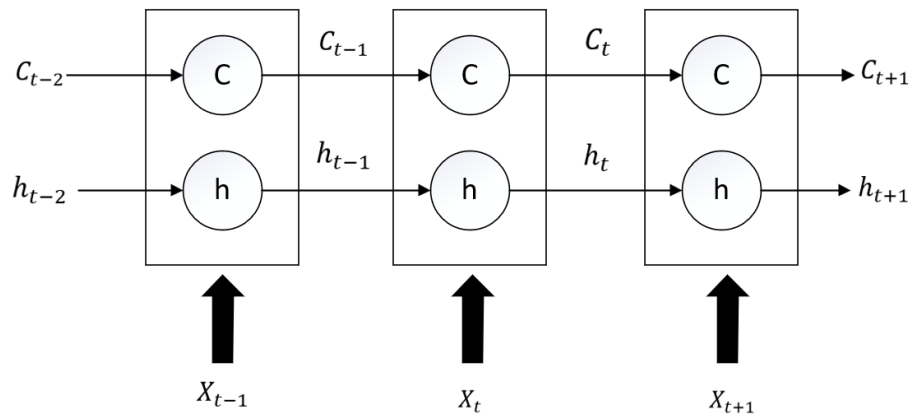
Imagen segmentada por U2-net. Fuente. Elaboración propia

**2.2.2 Predicción del crecimiento de la planta**

Para la predicción del crecimiento de la planta usaremos Deep Learning, específicamente un modelo LSTM, pues se ha observado que presenta buenos resultados para esta clase de tareas(Sakurai et al., n.d.). El modelo LSTM combina la estructura de una RNN, efectiva en series de tiempos cortos, y mejora su preservación a largo plazo agregando un estado de célula basado en la RNN(Wu et al., 2021). Su estructura expandida se puede apreciar en la Figura 11, donde X, h y c son vectores de entrada, salida y estado de célula respectivamente.

Figura 11

Vista expandida de red LSTM

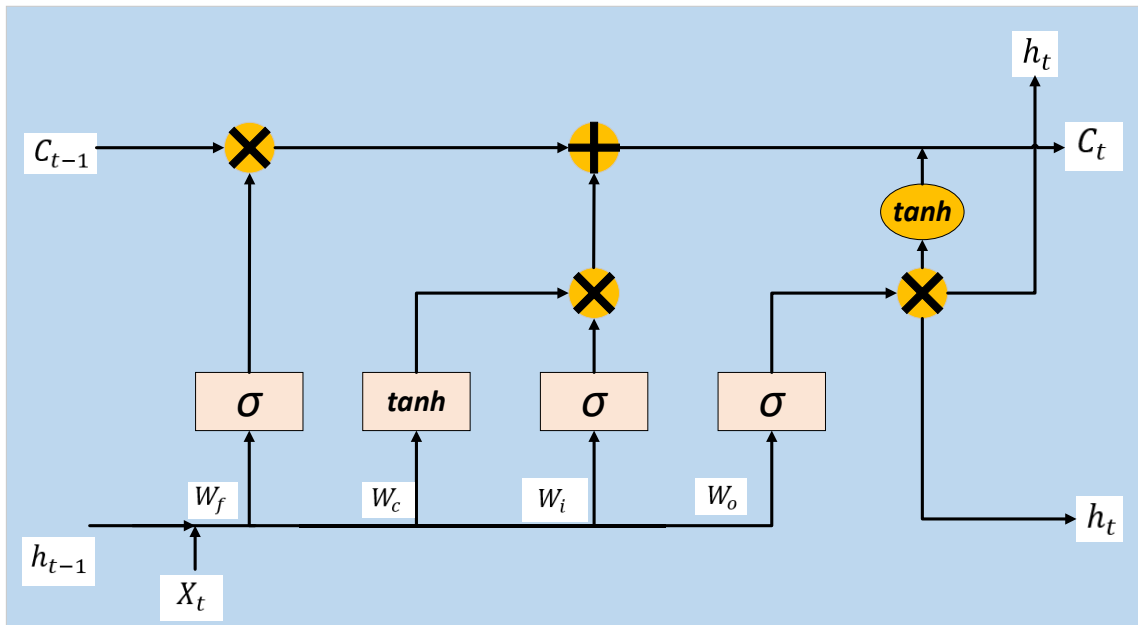


El modelo LSTM utiliza compuertas internas para gestionar la preservación, actualización y entrada del estado a largo plazo. Estas compuertas controlan la información almacenada en la celda y determinan qué partes del estado queremos incluir en la salida final. Su

estructura interna se observa en la Figura 12. Está compuesta por la compuerta del olvido, la compuerta de entrada, la compuerta de salida.

Figura 12

Arquitectura interna de la red LSTM. Fuente. Elaboración propia.



La compuerta del olvido, entrada y salida están descritas de la siguiente forma:

$$g(x) = \sigma(W_x + b) \tag{3}$$

Tal que:

g : es la compuerta

σ : es la función sigmoide

W : es la matriz de pesos de la compuerta

b : el vector de error

Siguiendo esta estructura, se representan las compuertas del olvido (f_t), entrada (i_t) y salida (o_t)

en función del tiempo:

$$f_t = \sigma(W_f * [h_{t-1}, X_t] + b_f) \quad (4)$$

$$i_t = \sigma(W_i * [h_{t-1}, X_t] + b_f) \quad (5)$$

$$o_t = \sigma(W_o * [h_{t-1}, X_t] + b_o) \quad (6)$$

Tal que:

h : es la salida del nodo de la capa oculta

x : es la entrada

Por otra parte, la unidad de estado de entrada c'_t en el tiempo actual t , se calcula a través de la salida de la red en tiempo $t - 1$ y la entrada en el tiempo t :

$$c'_t = \tanh(W_c[h_{t-1}, X_t] + b_c) \quad (7)$$

Para obtener el estado de la célula C_t en el tiempo actual t , se procede a la multiplicación de elemento por elemento:

$$C_t = f_t \circ C_{t-1} + i_t \circ c'_t \quad (8)$$

La salida de la red neuronal de memoria a largo plazo se determina de acuerdo con la puerta de salida y el estado de la célula:

$$h_t = o_t \circ \tanh(c_t) \quad (9)$$

Los hiperparámetros de la red LSTM como sus capas ocultas, neuronas y tasa de aprendizaje establecidos se pueden ver en la Tabla 4. Se escogieron en base a trabajos similares, sin embargo, se tuvieron que adaptar por limitaciones de cómputo.

Tabla 4

Hiperparámetros de la red LSTM propuesta.

Parámetro	Valor
Capas	150

Capas ocultas	0
Optimizador	Adam
Tasa de aprendizaje	0,01
Épocas	400
Tamaño de lote	1

3. Resultados

Para comparar los algoritmos de segmentación seleccionados se utilizó una planta 01-1 registrada en mayo de 2023. La planta se puede observar en la Figura 13.

Figura 13

Planta para 01-1 el 10 de mayo del 2023. Fuente. Elaboración propia.



3.1 PlantCV

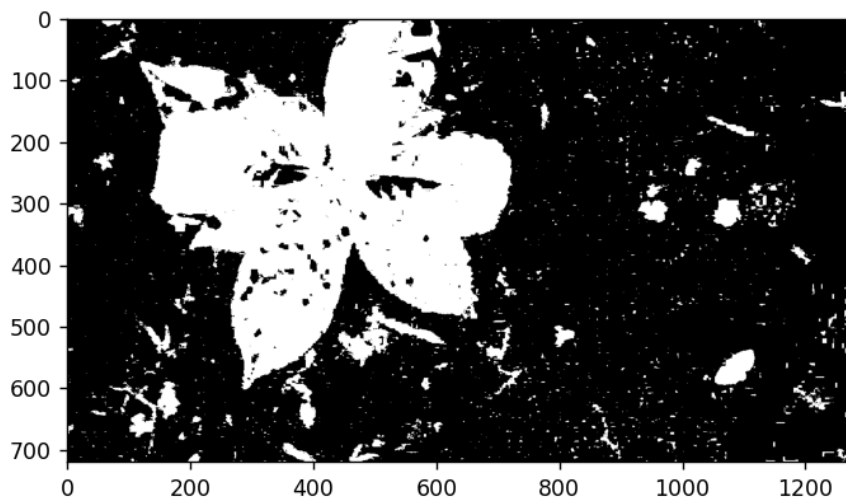
Al utilizar la librería PlantCV para segmentar nuestra base de datos se obtuvieron los siguientes resultados:

3.1.1 Segmentación dual HSV/LAB

Con el modelo HSV/LAB obtenemos una máscara con ruido en toda la imagen, esto se debe a la presencia de maleza alrededor de la planta, que presenta una gama de colores similar a ella. Este fenómeno se puede observar en la Figura 14.

Figura 14

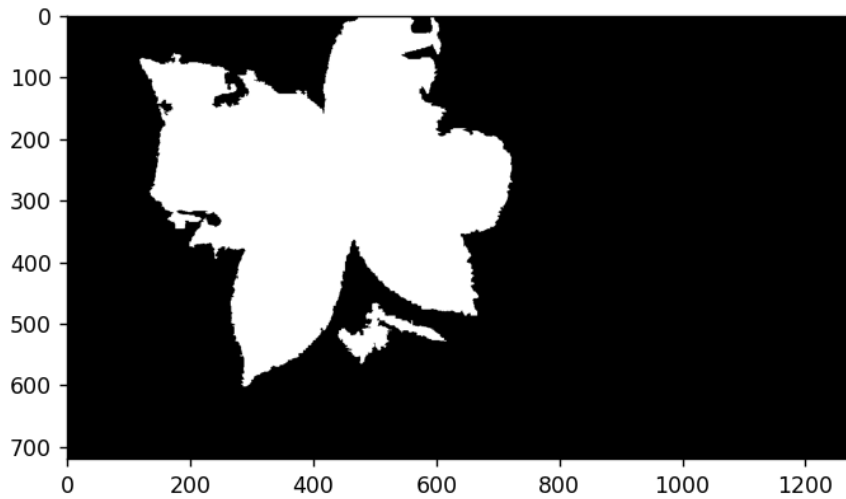
Máscara de Planta 01-1 el 10 de mayo del 2023 segmentada de forma dual. Fuente. Elaboración propia



Como se observa en la imagen presenta ciertas hojas incompletas y un poco de maleza que el algoritmo no pudo sustraer de la imagen correctamente. Con el fin de mejorar la imagen para la posterior predicción, se hace una limpieza por clusters k means por tamaños, donde se eliminan cuerpos que no cumplan con una dimensión establecida, además de rellenar intersticios dentro de la hoja. El resultado final se observa en la Figura 15.

Figura 15

Máscara final por segmentación dual por HSV/LAB. Fuente: Elaboración propia

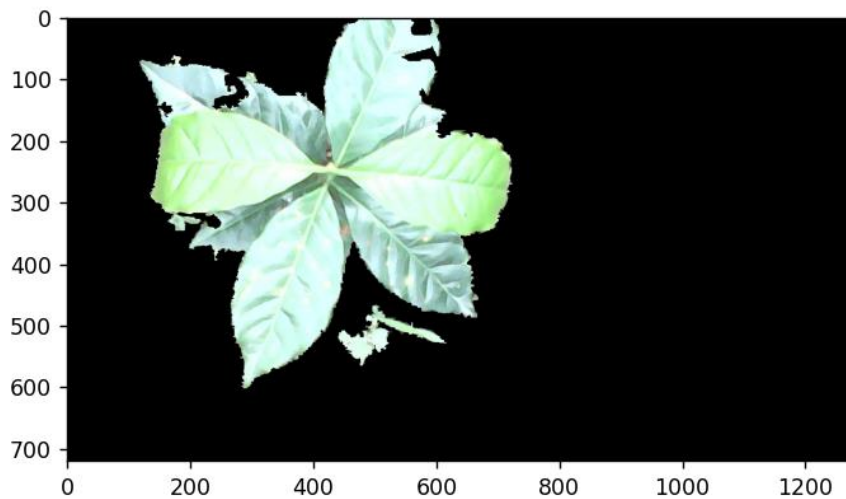


La imagen en formato RGB superpuesta con la máscara se puede observar en la Figura

16.

Figura 16

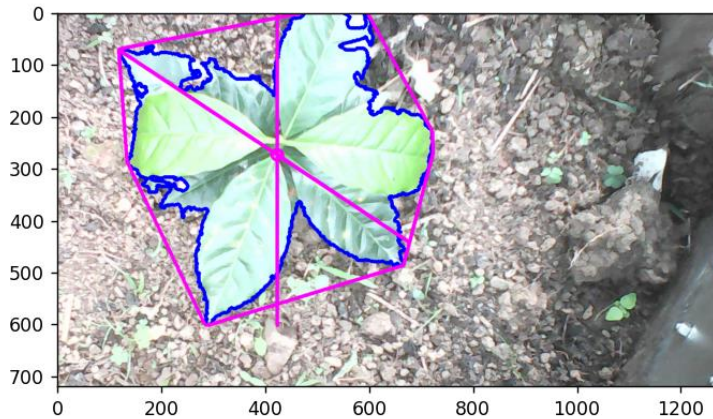
Planta 01-1 el 10 mayo del 2023 segmentada de forma dual. Fuente. Elaboración propia



Adicionalmente, se puede observar la morfología de la planta según la máscara creada en la Figura 17, de la cual se extrae información numérica de la planta, en este caso nos interesa el diámetro de la planta para generar la predicción.

Figura 17

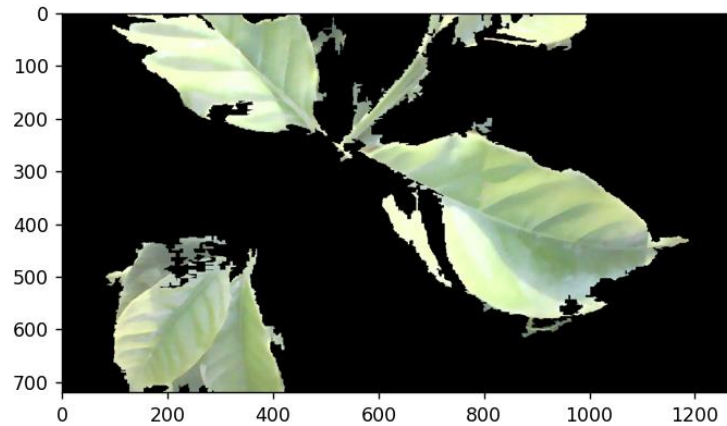
Morfología de la planta 01-1 el 10 de mayo de 2023. Fuente. Elaboración propia.



Sin embargo, el algoritmo no segmentó con la misma exactitud todas las imágenes, durante la ejecución de esta tarea de forma secuencial se obtuvieron imágenes donde la planta no se encontraba completa, creando segmentaciones de baja calidad, como se puede observar en la Figura 18.

Figura 18

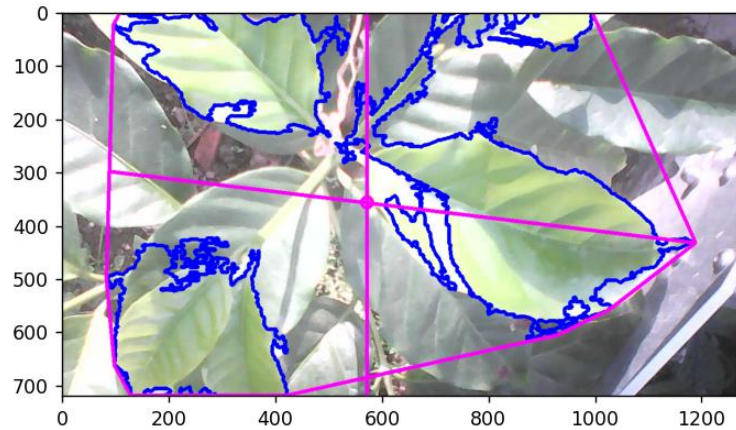
Planta 01-1 el 14 de octubre del 2023 segmentada de forma dual. Fuente. Elaboración propia



El algoritmo no reconoce por completo a la planta, posiblemente se debe a que la gama de colores haya cambiado por la presencia de sombras ocasionadas por el sol, esto se evidencia de mejor forma en la morfología de la planta en la Figura 19.

Figura 19

Morfología de la planta 01-1 el 14 octubre del 2023. Fuente. Elaboración propia.



Este tipo de segmentación es más preciso cuando las condiciones del entorno son controladas y estables, sin variaciones de luz que puedan afectar los colores. Este método no es funcional para esta clase de entornos si se espera automatizar.

3.1.2 Segmentación por Naive Bayes

Al ejecutar el algoritmo para segmentar las mismas dos imágenes se obtuvieron los resultados en la Figura 20 y Figura 21.

Figura 20

Planta 01-1 en mayo del 2023 segmentada por Naive Bayes. Fuente. Elaboración propia.

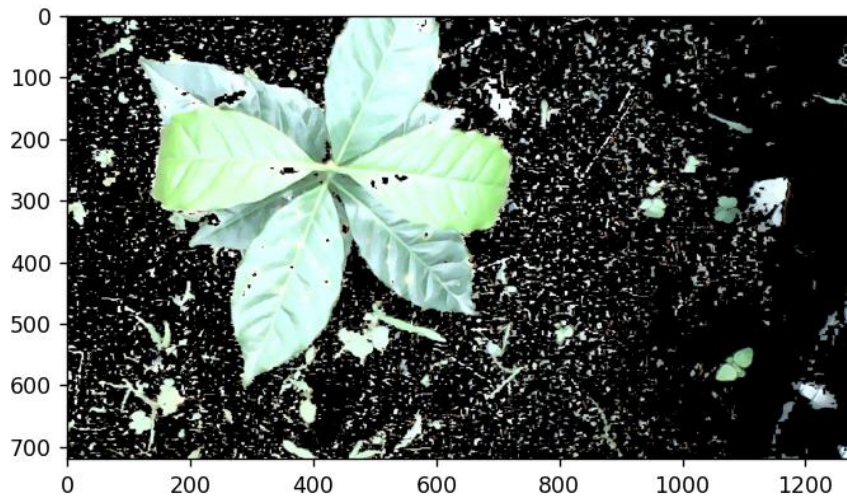
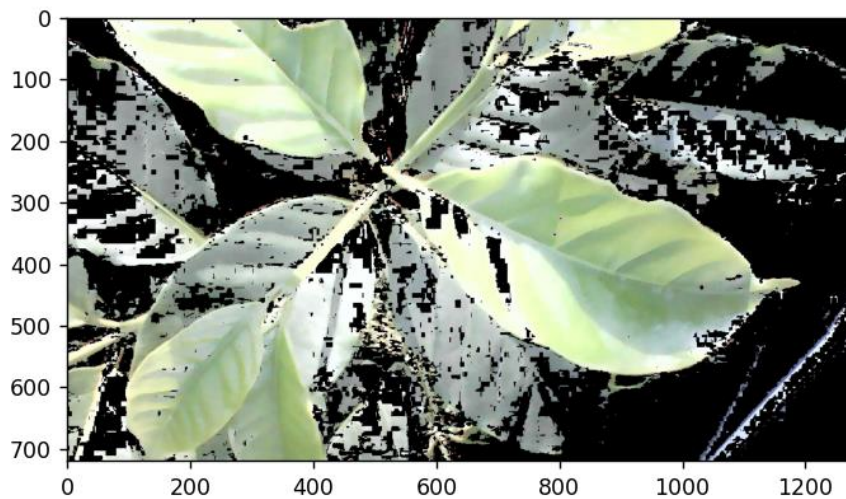


Figura 21

Planta 01-1 en octubre del 2023. Fuente. Elaboración propia.



El desarrollo de este algoritmo presenta resultados similares, esto se debe a que tanto la segmentación dual HSV/LAB como por Naive Bayes se desarrollan por medio de un análisis cromático del espectro visible, de ahí su similitud.

3.2 Red neuronal U2-net

La segmentación por redes neuronales de U2net tiene la capacidad de segmentar la planta de manera perfecta, pues obtiene únicamente el objeto más prominente en la imagen, que la mayor parte del tiempo corresponde a la planta como se puede observar en la Figura 22.

Figura 22

Planta 01-1 en mayo segmentada por U2net. Fuente. Elaboración propia.



Sin embargo, esto no ocurren en todos los casos, pues en algunas imágenes segmentadas la red neuronal detecta como cuerpo principal una sola hoja, y no es capaz de asociarla a las demás como se puede observar en la Figura 23.

Figura 23

Planta 01-1 en octubre. Fuente. Elaboración propia.



En la Figura 24 se puede observar la comparación de los resultados de la segmentación.

Figura 24

Comparación de segmentaciones: Naive Bayes, dual HSV/LAB y U2 net respectivamente.

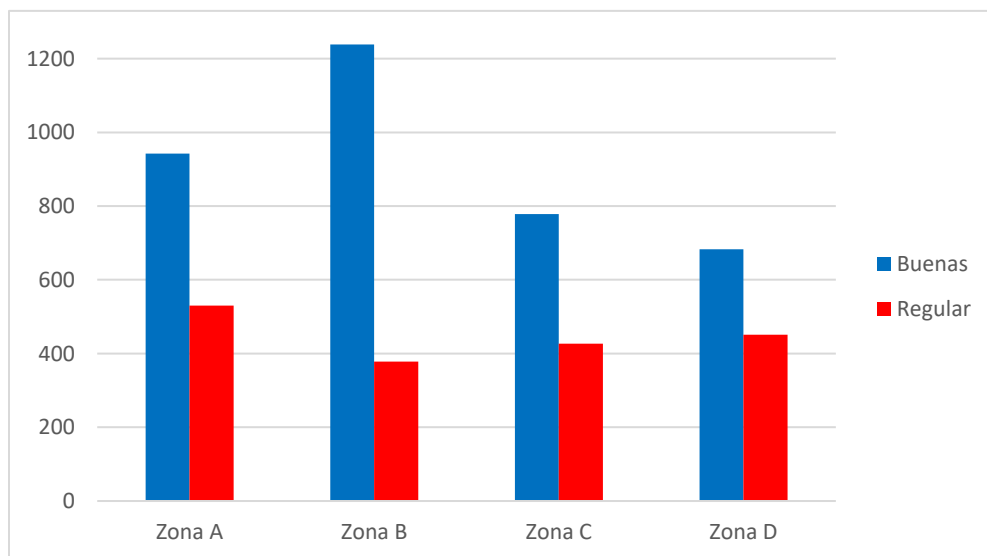
Fuente. Elaboración propia.



Se seleccionó la red neuronal U2-net como método para segmentar todo el dataset. Se obtuvo un 67% de imágenes segmentadas correspondientes a 3641 imágenes. Su distribución por zonas se puede observar en la Figura 25.

Figura 25

Resultados de segmentación de U2 net por zonas. Fuente. Elaboración propia.

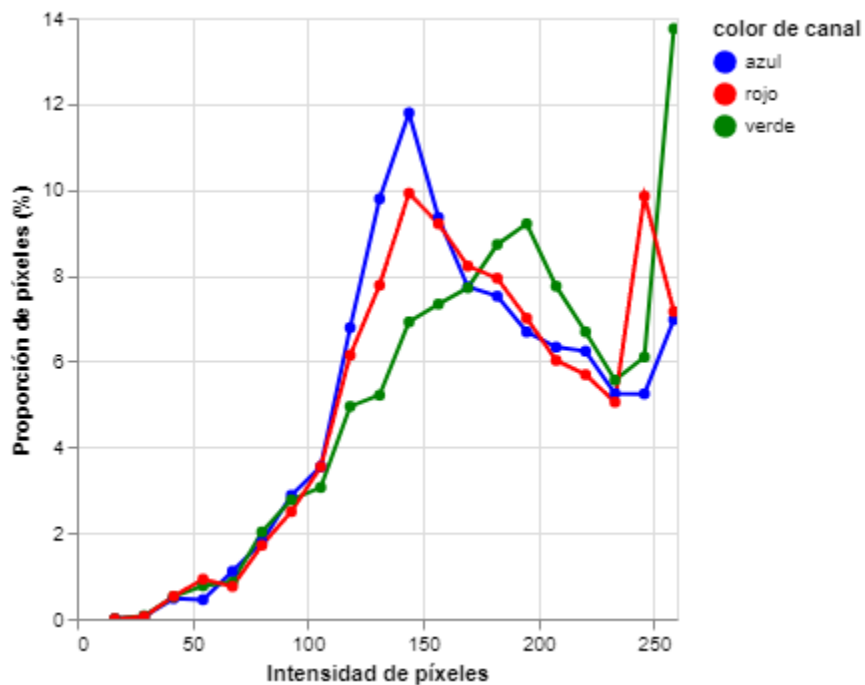


3.3 Análisis de espectro cromático en RGB

Se realizó un análisis del espectro cromático en RGB de las imágenes usando la librería PlantCV, se tomó un promedio entre las zonas y se graficó. Su resultado se puede ver en la Figura 26.

Figura 26

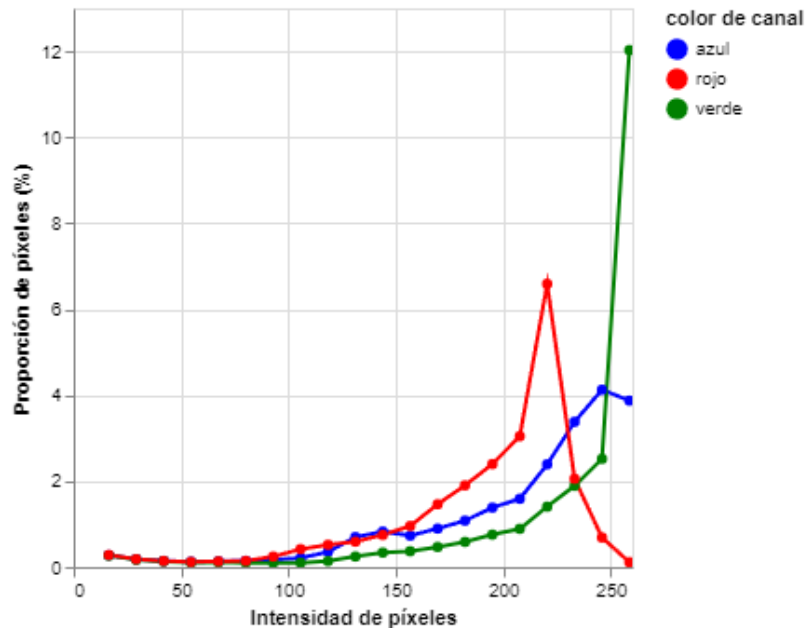
Análisis cromático en RGB de las imágenes sin segmentar. Fuente. Elaboración propia.



El eje x representa la intensidad de color al píxel, donde a cada píxel, se le asocia un valor que corresponde a su nivel de luminosidad en la imagen. El valor de esta magnitud representa la energía electromagnética capturada y se representa con un número positivo. Posteriormente se desarrolló el mismo análisis con el promedio de valores de las imágenes segmentadas, su resultado se puede observar en la Figura 27.

Figura 27

Análisis cromático en RGB de las imágenes segmentadas. Fuente. Elaboración propia.



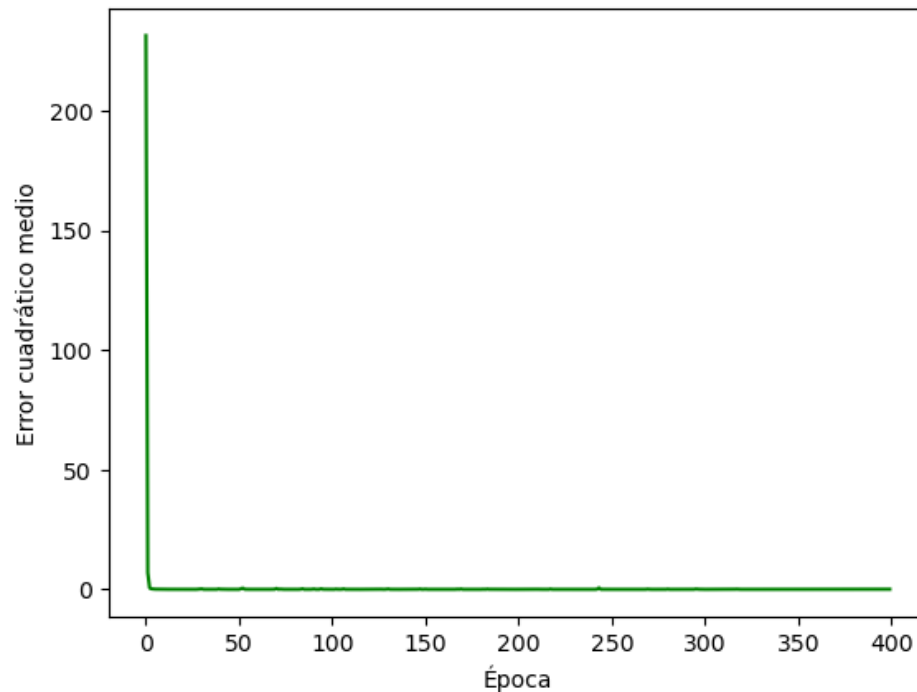
Resulta clara la disminución del canal azul y rojo respecto al verde, se observa la concentración de píxeles en valores superiores a 250 correspondientes al verde.

3.4 Predicción del crecimiento de la planta

Para la predicción del crecimiento de la planta se hizo una predicción del diámetro de la planta en cada una de las 4 zonas usando el promedio de las plantas por días y se comparó con la real, los hiperparámetros de entrenamiento se encuentran en la Tabla 4, se usaron el 80% de los datos como entrenamiento y el 20% como predicción. El error cuadrático medio del entrenamiento se puede observar en la Figura 28.

Figura 28

Error cuadrático medio vs número de épocas en entrenamiento. Fuente. Elaboración propia.



De la gráfica se puede observar que, al principio del entrenamiento, hay una disminución drástica del error cuadrático medio, lo que indica que el modelo está aprendiendo rápidamente a partir de los datos proporcionados. Después de la caída inicial, la línea se estabiliza, lo que sugiere que el modelo ha convergido. Con el fin de evaluar el algoritmo se ejerció una validación cruzada por k-fold, se dividió el conjunto de datos en 5 folds y se ejecutó el script. El resultado de cada fold se puede observar en la Tabla 5.

Tabla 5

Validación cruzada por k-fold.

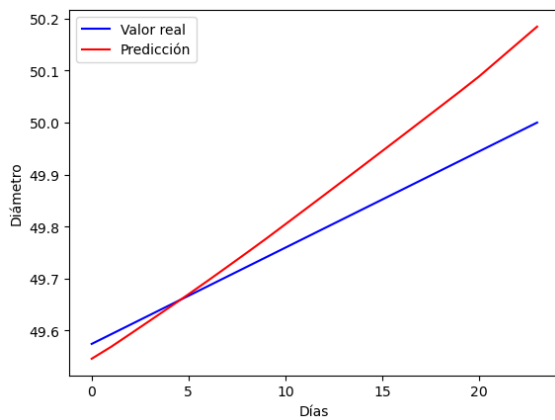
k-fold	Pérdida
1	4,9625
2	0,5442

3	0,2692
4	0,0097
5	0,0020

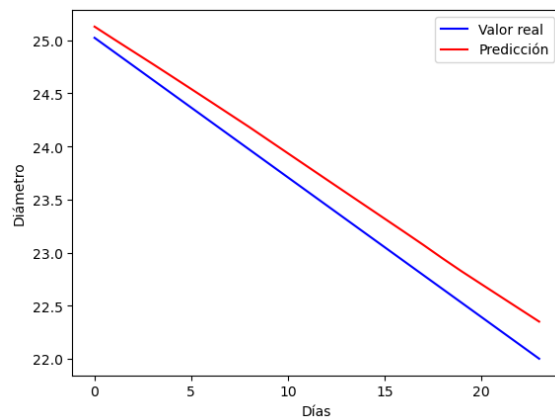
Hay una notable disminución en la pérdida desde el primer hasta el quinto pliego. Esto indica que el modelo se está ajustando a medida que tiene acceso a más datos e identifica más adecuadamente los patrones en cada iteración.

Figura 29

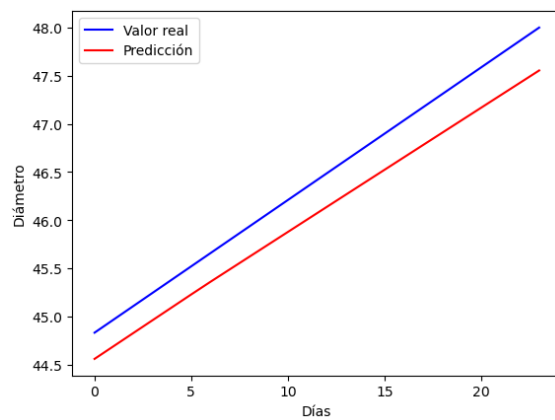
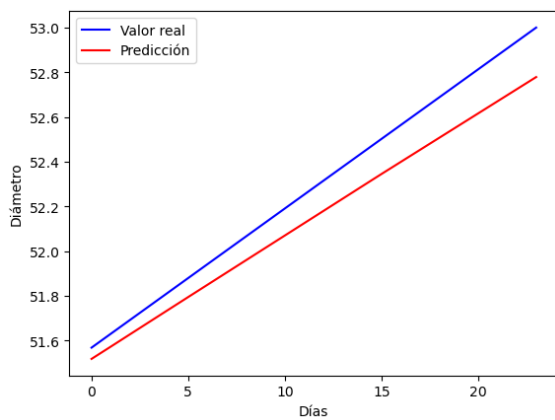
Predicción de crecimiento del diámetro de la planta por zonas. (a)Zona A, (b)Zona B, (c)Zona C, (d)Zona D. Fuente. Elaboración propia.



(a)



(b)



(c)

(d)

De la Figura 29 se aprecia que tanto los valores reales como las predicciones parecen seguir una la misma tendencia a lo largo del tiempo. Esto indica que el modelo ha aprendido correctamente el comportamiento del crecimiento de la planta además de su cercanía con la función real. La zona C fue la que obtuvo un mayor crecimiento de las 4 zonas, mostrando que la mejor forma de gestionar el crecimiento de las plantas es únicamente usando abono y sin regarle agua, adicional a las lluvias. La Zona B mostró un diámetro decreciente, significando que a las plantas se estaban marchitando lo que refleja una equivalencia con las imágenes.

4. Conclusiones

Nuestra evaluación comparativa de varios modelos de segmentación ha destacado la superioridad del enfoque basado en U2-net en comparación con los métodos de Naive Bayes y HSV/LAB. Esta ventaja es particularmente notable en términos de precisión y eficiencia en la clasificación en diversas condiciones de iluminación y entorno.

Se estableció que la metodología de segmentación propuesta, especialmente el modelo U2-net, es altamente efectiva para la identificación precisa de *Coffea arabica* en entornos complejos. Esto fue confirmado por una precisión de segmentación superior al 67%, lo que sugiere una base sólida para futuras aplicaciones de monitoreo automático de cultivos.

Se logró demostrar que el crecimiento de la planta *Coffea arabica* puede ser parametrizado y caracterizado para predecir su crecimiento en días futuros, esto, con el fin de seleccionar la mejor zona con sus variables de crecimiento. Los resultados de la validación cruzada en los últimos pliegues indican una excelente capacidad de predicción del algoritmo,

obteniendo una pérdida de solo 0.002 en la última iteración. Además, la disminución constante de la pérdida sugiere que el modelo no se sobreajustó a los datos de entrenamiento.

Se recomienda la expansión de la base de datos de imágenes y la exploración de redes neuronales convolucionales más profundas para mejorar aún más la precisión de las predicciones de crecimiento de las plantas. Además, la integración de datos ambientales y climáticos en el modelo podría proporcionar predicciones más robustas y adaptativas.

Referencias Bibliográficas

- Abicht, J., Hellmich, A., Wiese, T., Harst, S., & Ihlenfeldt, S. (2024). New automation solution for brownfield production – Cognitive robots for the emulation of operator capabilities. *CIRP Journal of Manufacturing Science and Technology*, *50*, 104–112.
<https://doi.org/10.1016/J.CIRPJ.2024.02.007>
- Abuhayi, B. M., & Mossa, A. A. (2023). Coffee disease classification using Convolutional Neural Network based on feature concatenation. *Informatics in Medicine Unlocked*, *39*, 101245. <https://doi.org/10.1016/j.imu.2023.101245>
- Ahmad, H. A., Novianti Winarlie, A., & Miranda, E. (2022). Indonesia Covid-19 Pandemic Social Media Analysis With Text Mining. *2022 International Conference on Information Management and Technology (ICIMTech)*, 94–99.
<https://doi.org/10.1109/ICIMTech55957.2022.9915051>
- Berry, J. C., Fahlgren, N., Pokorny, A. A., Bart, R. S., & Veley, K. M. (2018). An automated, high-throughput method for standardizing image color profiles to improve image-based plant phenotyping. *PeerJ*, *6*, e5727. <https://doi.org/10.7717/peerj.5727>
- Cui, Z., Huang, S., Wang, M., Nie, K., Fang, Y., & Tan, T. (2021). Improving the CFPP property of biodiesel via composition design: An intelligent raw material selection strategy based on different machine learning algorithms. *Renewable Energy*, *170*, 354–363.
<https://doi.org/10.1016/J.RENENE.2021.02.008>
- Didier, N. (2024). Turning fragments into a lens: Technological change, industrial revolutions, and labor. *Technology in Society*, *77*, 102497.
<https://doi.org/10.1016/J.TECHSOC.2024.102497>

Federación Nacional de Cafeteros de Colombia. (2023). *Café de Santander*.

<https://santander.federaciondefcafeteros.org/cafe-de-santander>.

Hunter, M. C., Smith, R. G., Schipanski, M. E., Atwood, L. W., & Mortensen, D. A. (2017).

Agriculture in 2050: Recalibrating Targets for Sustainable Intensification. *BioScience*, 67(4), 386–391. <https://doi.org/10.1093/biosci/bix010>

IBM. (2023). *What is deep learning?* <https://www.ibm.com/topics/deep-learning>

Kusiak, A. (2018). Smart manufacturing. *International Journal of Production Research*, 56(1), 508–517. <https://doi.org/10.1080/00207543.2017.1351644>

Lee, C.-J., Yang, M.-D., Tseng, H.-H., Hsu, Y.-C., Sung, Y., & Chen, W.-L. (2023). Single-plant broccoli growth monitoring using deep learning with UAV imagery. *Computers and Electronics in Agriculture*, 207. <https://doi.org/10.1016/j.compag.2023.107739>

Leng, J., Zhu, X., Huang, Z., Li, X., Zheng, P., Zhou, X., Mourtzis, D., Wang, B., Qi, Q., Shao, H., Wan, J., Chen, X., Wang, L., & Liu, Q. (2024). Unlocking the power of industrial artificial intelligence towards Industry 5.0: Insights, pathways, and challenges. *Journal of Manufacturing Systems*, 73, 349–363. <https://doi.org/10.1016/J.JMSY.2024.02.010>

Madakam, S., Uchiya, T., Madakam, S., & Uchiya, T. (2019). Industrial Internet of Things (IIoT): Principles, Processes and Protocols. *Computer Communications*, 35–53.

https://doi.org/10.1007/978-3-030-24892-5_2

Nugraha, F. S., Rachmat, H., Eka Atmaja, D. S., & Astuti, M. D. (2020). Application Design of Farmbot Mobile App Using Internet of Things on Android. *IOP Conference Series: Materials Science and Engineering*, 982(1), 012055. <https://doi.org/10.1088/1757-899X/982/1/012055>

- Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., & Jagersand, M. (2020). U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition*, *106*, 107404. <https://doi.org/10.1016/J.PATCOG.2020.107404>
- Quan, N. M., Phung, H. M., Uyen, L., Dat, L. Q., Ngoc, L. G., Hoang, N. M., Tu, T. K. M., Dung, N. H., Ai, C. T. D., & Trinh, D. N. T. (2023). Species and geographical origin authenticity of green coffee beans using UV–VIS spectroscopy and PLS–DA prediction model. *Food Chemistry Advances*, *2*, 100281. <https://doi.org/10.1016/j.focha.2023.100281>
- Sakurai, S., Uchiyama, H., Shimada, A., & Taniguchi, R.-I. (n.d.). *Plant Growth Prediction using Convolutional LSTM*. <https://doi.org/10.5220/0007404901050113>
- Sano, A. V. D., Stefanus, A. A., Madyatmadja, E. D., Nindito, H., Purnomo, A., & Sianipar, C. P. M. (2023). Proposing a visualized comparative review analysis model on tourism domain using Naïve Bayes classifier. *Procedia Computer Science*, *227*, 482–489. <https://doi.org/10.1016/j.procs.2023.10.549>
- SgROI, F. (2023). Precision agriculture and competitive advantage: Economic efficiency of the mechanized harvesting of Chardonnay and Nero d’Avola grapes. *Journal of Agriculture and Food Research*, *14*, 100774. <https://doi.org/10.1016/J.JAFR.2023.100774>
- Sundari, V. K., Nithyashri, J., Kuzhaloli, S., Subburaj, J., Vijayakumar, P., & Jose, P. S. H. (2021). Comparison analysis of IoT based industrial automation and improvement of different processes – review. *Materials Today: Proceedings*, *45*, 2595–2598. <https://doi.org/10.1016/J.MATPR.2020.11.338>
- Wu, X., Liu, Z., Yin, L., Zheng, W., Song, L., Tian, J., Yang, B., & Liu, S. (2021). A Haze Prediction Model in Chengdu Based on LSTM. *Atmosphere 2021*, Vol. 12, Page 1479, *12*(11), 1479. <https://doi.org/10.3390/ATMOS12111479>

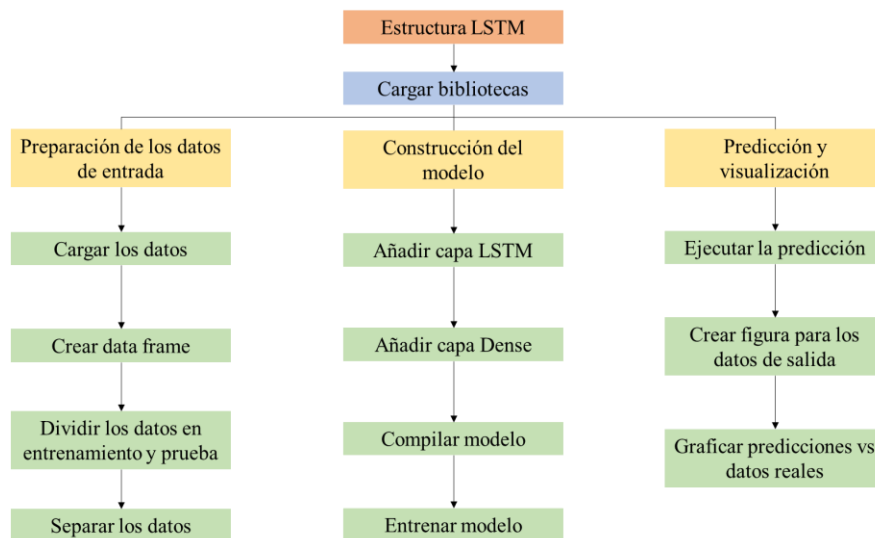
Anexos

Los códigos de segmentación y predicción se encuentran anexados en el siguiente link de github: https://github.com/hernansanchez34/Plant_growth_prediction

Estructura del código de predicción

La Figura 30 a continuación muestra la estructura del código implementado en la red LSTM

Figura 30. Esquema estructural del código.



1. Se importan las librerías numpy, pandas, matplotlib, openpyxl y keras.
2. Se carga el archivo donde se encuentren los datos usando openpyxl, en este caso fue un .xlsx de Excel.
3. Se crea una lista vacía para posteriormente crear el data frame.
4. Se itera para anexar cada uno de los datos del archivo Excel a la lista.
5. Se selecciona las columnas que se van a leer del data frame creado[Día, Abono, Agua, Diámetro].
6. Se separan lo datos datos de entrenamiento en 80/20, entrenamiento y predicción respectivamente.

7. Se hace un reshape de los datos de entrenamiento y predicción para que el modelo de keras se pueda crear.
8. Se crea el modelo inicializándolo por la función de keras sequential y sus respectivas capas de acuerdo con los hiperparámetros. Ver Tabla 4. La función de activación que mejor respondió al entrenamiento fue RELU.
9. Se entrena el modelo con los datos correspondientes y se ejecuta la predicción.
10. Se grafica predicción y crecimiento real vs días.

Para evaluar la mejor configuración de entrenamiento se utilizaron 3 métricas el error cuadrático medio, el error porcentual absoluto medio y el error absoluto medio, ecuación 10, 11, 12.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (10)$$

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{(y_i - \hat{y}_i)^2}{y_i} \right| \quad (11)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (12)$$

Tal que:

y_i es el valor real.

\hat{y}_i es el valor predicho por el modelo.

n es el número de datos.

Teniendo estas 3 métricas en cuenta se variaron los hiperparámetros para encontrar el mejor entrenamiento y se calculó una magnitud teniendo en cuenta el valor de MSE MAPE y MAE ver ecuación 13, ver Tabla 6, Tabla 7 y Tabla 8.

$$Magnitud = \sqrt{MSE^2 + MAE^2 + MAPE^2} \quad (13)$$

Tabla 6

Iteración por épocas del entrenamiento.

Neuronas	Epocas	Tamaño del lote	Optimizador	Función de activación	MSE	MAPE	MAE	Magnitud
150	400	1	ADAM	Relu	8.98E-03	2.89E-01	7.44E-02	0.298
150	200	1	ADAM	Relu	5.21E-01	1.70E+00	5.62E-01	1.866
150	100	1	ADAM	Relu	2.15E-02	1.06E-01	5.58E-01	0.568
150	50	1	ADAM	Relu	8.35E-02	6.35E-01	2.08E-01	0.674
150	25	1	ADAM	Relu	2.39E-02	1.25E-01	7.10E-01	0.721

Tabla 7

Iteración por función de activación y tamaño de lote en entrenamiento.

Neuronas	Epocas	Tamaño del lote	Optimizador	Función de activación	MSE	MAPE	MAE	Magnitud
150	400	1	ADAM	Relu	8.98E-03	2.89E-01	7.44E-02	0.298
150	400	2	ADAM	Relu	3.47E-02	5.65E-01	1.50E-01	0.585
150	400	1	ADAM	Sigmoid	7.09E-02	8.68E-01	1.94E-01	0.892
150	400	2	ADAM	Sigmoid	5.40E-02	9.09E-01	1.74E-01	0.927
150	400	1	ADAM	Tanh	7.40E-02	8.14E-01	2.04E-01	0.843
150	400	2	ADAM	Tanh	1.18E-01	9.25E-01	2.77E-01	0.973

Tabla 8

Iteración por número de neuronas en entrenamiento.

Neuronas	Epocas	Tamaño del lote	Optimizador	Función de activación	MSE	MAPE	MAE	Magnitud
150	400	1	ADAM	Relu	8.98E-03	2.89E-01	7.44E-02	0.298
100	400	1	ADAM	Relu	2.93E-02	4.38E-01	1.24E-01	0.456
50	400	1	ADAM	Relu	2.38E-02	3.93E-01	1.11E-01	0.409
25	400	1	ADAM	Relu	3.86E-02	4.28E-01	1.41E-01	0.452

De esta forma se encontraron los mejores hiperparámetros y su función de activación(Relu), como se puede ver en la Tabla 4.